PATENT APPLICATION

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

IN RE APPLICATION OF:	ATTORNEY'S DOCKET NUMBER
MARK KANTROWITZ	**2942-991842**

ENTITLED

"Method And Apparatus For Efficient Identification Of Duplicate And Near-Duplicate Documents And Text Spans Using High-Discriminability Text Fragments"

BOX PATENT APPLICATION
Assistant Commissioner for Patents
Washington, D.C. 20231

EXPRESS MAIL CERTIFICATE

"Express Mail" Label Number EL561512695US

Date of Deposit November 15, 2000

I hereby certify that the following attached papers or fee

UTILITY PATENT APPLICATION TRANSMITTAL (1 p.); FEE TRANSMITTAL FOR FY 2000 (1 p.); SPECIFICATION (12 pp.); CLAIMS (6 pp., 39 claims); ABSTRACT (1 p.); NINE SHEET OF DRAWINGS (Figs. 1-6); DECLARATION AND POWER OF ATTORNEY (2 pp.); RECORDATION FORM COVER SHEET - PATENTS ONLY (2 pp.); ASSIGNMENT (2 pp.); and two checks in amounts of $1,052.00 and $40.00

is being deposited with the United States Postal Service "Express Mail Post Office to Addressee" service under 37 C.F.R. §1.10 on the date indicated above and is addressed to the Assistant Commissioner for Patents, Washington, D.C. 20231.

Linda L. Marlowe
(Typed name of person mailing paper or fee)

Linda L. Marlowe
(Signature of person mailing paper or fee)

METHOD AND APPARATUS FOR EFFICIENT IDENTIFICATION OF DUPLICATE AND NEAR-DUPLICATE DOCUMENTS AND TEXT SPANS USING HIGH-DISCRIMINABILITY TEXT FRAGMENTS

BACKGROUND OF THE INVENTION

1. Field of the Invention

This invention relates to a computer-assisted method and apparatus for identifying duplicate and near-duplicate documents or text spans in a collection of documents or text spans, respectively.

2. Description of the Prior Art

The current art includes inventions that compare a single pair of known-to-be-similar documents to identify the differences between the documents. For example, the Unix "diff" program uses an efficient algorithm for finding the longest common sub-sequence (LCS) between two sequences, such as the lines in two documents. Aho, Hopcroft, and Ullman, *Data Structures and Algorithms, Addison-Wesley Publishing Company, April 1987, pages 189-192*. The lines that are left when the LCS is removed represent the changes needed to transform one document into another. Additionally, U.S. Patent No. 4,807,182 uses anchor points (points in common between two files) to identify differences between an original and a modified version of a document. There are also programs for comparing a pair of files, such as the Unix "cmp" program.

Another approach for comparing documents is to compute a checksum for each document. If two documents have the same checksum, they are likely to be identical. But comparing documents using checksums is an extremely fragile method, since even a single character change in a document yields a different checksum. Thus, checksums are good for identifying exact duplicates, but not for identifying near-duplicates. U.S. Patent No. 5,680,611 teaches the use of checksums to identify duplicate records. U.S. Patent No. 5,898,836 discloses the use of checksums to identify whether a region of a document has changed by comparing checksums for sub-document passages, for example, the text between HTML tags.

Patrick Juola's method, discussed in Juola, Patrick, *What Can We Do With Small Corpora? Document Categorization via Cross-Entropy*, Proceedings of Workshop on Similarity and Categorization, 1997, uses the average length of matching character n-grams (an n-gram is a string of characters that may comprise all or part of a word) to identify similar documents. For each window of consecutive characters in the source document, the average length of the longest matching sub-sequence at each position in the target document is computed. This effectively computes the average length of match at each position within the target document (counting the number of consecutive matching characters starting from the first character of the n-gram) for every possible character n-gram within the source document. This technique depends on the frequency of the n-grams within the document by requiring the n-grams and all sub-parts (at least the prefix sub-parts) to be of high frequency. The Juola method focuses on applications involving very small training corpora, and has been applied to a variety of areas, including language identification, determining authorship of a document, and text classification. The method does not provide a measure of distinctiveness.

The prior art does not compare more than two documents, does not allow text fragments in each document to appear in a different or arbitrary order, is not selective in the choice of n-grams used to compare the documents, does not use the frequency of the n-grams across documents for selecting n-grams used to compare the documents, and does not permit a mixture of very low frequency and very high frequency components in the n-grams.

SUMMARY OF THE INVENTION

It is an object of the present invention to provide a method and apparatus for the efficient identification of duplicate and near-duplicate documents and text spans.

Accordingly, I have developed a method and apparatus for the efficient identification of duplicate and near-duplicate (i.e., substantially duplicate) documents and text spans which use high-discriminability text fragments for comparing documents.

Near-duplicate documents contain long stretches of identical text that are not present in other, non-duplicate documents. The long text fragments that are present in only a few documents (high-intermediate rarity) represent distinctive features that can be used to distinguish similar documents from dissimilar documents in a robust fashion. These text fragments represent a kind of "signature" for a document which can be used to match the document with near-duplicate documents and to distinguish the document from non-duplicate documents. Documents that overlap significantly on such text fragments will most likely be duplicates or near-duplicates. Overlap occurs not just when the text is excerpted, but also when deliberate changes have been made to the text, such as paraphrasing, interspersing comments by another author, and outright plagiarism.

Typically, as long as the document is not completely rewritten, there will be large text fragments that are specific to the document and its duplicates. On the other hand, text fragments in common between two non-duplicate documents will likely be in common with many other documents.

The present invention identifies duplicate and near-duplicate documents and text spans by identifying a small number of distinctive features for each document, for example, distinctive word n-grams likely to appear in duplicate or near-duplicate documents. The features act as a proxy for the full document, allowing the invention to compare documents by comparing their distinctive features. Documents having at least one feature in common are compared with each other. Near-duplicate documents are identified by counting the proportion of the features in common between the two documents. Using these common features allows the present invention to find near-duplicate documents efficiently without needing to compare each document with all the other documents in the collection, for example, by pairwise comparison. By comparing features instead of entire documents, the present invention is much faster in finding duplicate and near-duplicate documents in a large collection of documents than might be possible with prior document comparison algorithms.

A key to the effectiveness of this method is the ability to find distinctive features. The features need to be rare enough to be common among only near-duplicate documents, but not so rare as to be specific to just one document. An individual word may not be rare enough, but an n-gram containing the word might be. Longer n-grams might be too rare. Additionally, the distinctive features may include glue words (i.e., very common words) within the features but, preferably, not at either end. Thus, distinctive features may include words that are common to just a few documents and/or words that are common to all but a few documents.

Blindly gathering all n-grams of appropriate rarity would yield a computationally expensive algorithm. Thus, the number of distinctive features used must be small in order for the algorithm to be computationally efficient. The present invention incorporates several methods that strike a balance between appropriate rarity and computational expense.

Applications of the present invention include removing redundancy in document collections (including web catalogs and search engines), matching summary sentences with corresponding document sentences, and detection of plagiarism and copyright infringement for text documents and passages.

BRIEF DESCRIPTION OF THE DRAWINGS

Fig. 1 is a flow diagram of a first embodiment of a method according to the present invention as applied to documents;

Fig. 2 is a flow diagram of a second embodiment of a method according to the present invention as applied to documents;

Figs. 3A and 3B are a flow diagram of a third embodiment of a method according to the present invention as applied to documents;

Fig. 3C is an illustration of a document index;

Fig. 3D is an illustration of a feature index;

Fig. 3E is an illustration of a list 324;

Fig. 3F is an illustration of a list 330;

Fig. 3G is an illustration of a list 336;

Fig. 4 is a flow diagram of an embodiment of a method according to the present invention as applied to text spans;

Fig. 5 is a flow diagram of an embodiment of a method according to the present invention as applied to images; and

Fig. 6 is an illustration of an apparatus according to the present invention.

DESCRIPTION OF THE PREFERRED EMBODIMENTS

Referring to Fig. 1, the present invention is utilized to find duplicate or near-duplicate documents within a document collection 100. Step 110 identifies distinctive features in the document collection 100 and in each document in the collection 100. Loop 112 iterates for each pair of documents. Within loop 112, step 114 determines if the pair of documents has at least one distinctive feature in common. If they do, the pair is compared in step 116 to determine if they are duplicate or near-duplicate documents. Loop 112 then continues with the next pair of documents. If the pair of documents does not have at least one distinctive feature in common, no comparison is performed, and loop 112 continues with the next pair of documents.

The method illustrated in Fig. 1 can be applied to, for example: removing duplicates in document collections; detecting plagiarism; detecting copyright infringement; determining the authorship of a document; clustering successive versions of a document from among a collection of documents; seeding a text classification or text clustering algorithm with sets of duplicate or near-duplicate documents; matching an e-mail message with responses to the e-mail message, and vice versa; and creating a document index for use with a query system to efficiently find documents that contain a particular phrase or excerpt in response to a query, even if the particular phrase or excerpt was not recorded correctly in the document or the query.

The method can also be applied to augmenting information retrieval or text classification algorithms that use single-word terms with a small number of multi-word terms. Algorithms of this type that are based on a bag-of-words model assume that each word appears independently. Although such algorithms can be extended to apply to word bigrams, trigrams, and so on, allowing all word n-grams of a particular length rapidly becomes computationally unmanageable. The present invention may be used to generate a small list of word n-grams to augment the bag-of-words index. These word n-grams are likely to distinguish documents. Therefore, if they are present in a query, they can help narrow the search results considerably. This is in contrast to methods based on word co-occurrence statistics which yield word n-grams that are rather common in the document set.

The method illustrated in Fig. 1 may be used to determine whether documents are duplicates or near-duplicates even if the distinctive features appear in a different order in each document.

The distinctive features may be distinctive text fragments found within the collection of documents 100. As such, the method may be applied to information retrieval methods, such as a text classification method or any information retrieval method that assumes word independence and adds the distinctive text fragments to an index set.

The distinctive text fragments may be sequences of at least two words that appear in a limited number of documents in the document collection 100. If one distinctive text fragment is found within another distinctive text fragment, only the longest distinctive text fragment may be considered as a feature. A sequence of at least two words may be considered as appearing in a document when the document contains the sequence of at least two words at least a user-specified minimum number of times or a user-specified minimum frequency. The frequency may be defined as the number of occurrences in the document divided by the length of the document.

For each sequence of at least two words, a distinctiveness score may be calculated and the highest scoring sequences that are found in at least two documents in the document collection 100 may be considered as text fragments. The distinctiveness score may be the reciprocal of the number of documents containing the phrase multiplied by a monotonic function of the number of words in the phrase, where the monotonic function may be the number of words in the phrase.

The limited number restricting the number of documents having the sequence of at least two words may be selected by a user as a constant or a percentage. The limited number may be defined by a linear function of the number of documents in the document collection 100, such as a linear function of the square root or logarithm of the number of documents in the document collection 100.

The distinctive text fragments may include glue words (i.e., words that appear in almost all of the documents and for which their absence is distinctive). Glue words include stopwords like "the" and "of" and allow phrases like "United States of America" to be counted as distinctive phrases. The method may exclude glue words that appear at either extreme of the distinctive text fragment. Again, the sequence of at least two words may be considered as appearing in a document when the document contains the sequence of at least two words at least a user-specified minimum number of times or a user-specified minimum frequency. The frequency may be defined as the number of occurrences in the document divided by the length of the document.

Fig. 2 illustrates another embodiment of the present invention which finds duplicate or near-duplicate documents within a document collection 200. Step 210 identifies distinctive features of the documents in the document collection 200 and in each document in the collection 200. Loop 212 iterates for each pair of documents. Within loop 212, step 214 determines if the pair of documents has at least one distinctive feature in common. If they do, step 216 divides the number of features that the pair of documents has in common by the smaller number of the number of features in each document. Step 218 determines whether the result of step 216 is greater than a threshold value. The threshold value may be a constant, a fixed percentage of the number of documents in the document collection 200, the logarithm of the number of documents, or the square root of the number of documents. If the result is greater than the threshold, step 220 deems the documents duplicates or near-duplicates, and loop 212 continues with the next pair of documents. If the result is not greater than the threshold, the documents are not duplicates or near-duplicates, and loop 212 continues.

Figs. 3A and 3B show another embodiment of the present invention which finds duplicate or near-duplicate documents within a document collection 300. Starting with Fig. 3A, step 310 identifies distinctive features of the documents in the document collection 300 and in each document in the collection 300. Step 312 builds a document index 314 and step 316 builds a feature index 318. The document index 314 maps each document to the features contained therein. The feature index 318 maps the features to the documents that contain them. The indexes 314 and 318 are built in a manner that ignores duplicates (i.e., if a feature is repeated within a document, it is mapped only once). Loop 320 iterates through each document such that step 322 can create a list 324 that includes each unique distinctive feature that was identified in step 310. For each distinctive feature in list 324, loop 326 iterates through the feature index 318 so that step 328 can create a list 330 that includes each distinctive feature and the documents in which the distinctive feature is located.

Referring now to Fig. 3B, loop 332 iterates through list 330. Within loop 332, step 334 creates a list 336 of pairs of documents that have at least one feature in common and the number of features they have in common. Loop 338 iterates through list 336. For each pair of documents in list 336, step 340 divides the number of features that the pair of documents has in common by the smaller number of the number of features in each document (from the document index 314). Step 342 determines whether the result of step 340 is greater than a threshold value. The threshold value may, for example, be a constant, a fixed percentage of the number of documents in the document collection 300, the logarithm of the number of documents, or the square root of the number of documents. If the result is greater than the threshold, step 344 deems the documents duplicates or near-duplicates, and loop 338 continues with the next pair of documents. If the result is not greater than the threshold, the documents are not duplicates or near-duplicates, and loop 338 continues.

Fig. 3C illustrates an example format for the document index 314. Likewise, Fig. 3D illustrates the feature index 318, Fig. 3E illustrates list 324, Fig. 3F illustrates list 330, and Fig. 3G illustrates list 336 as constructed in two steps.

Referring to Fig. 4, a method according to the present invention is utilized to find duplicate or near-duplicate text spans, including sentences, within a text span collection 400. The text spans in the collection 400 may be sentences. Step 410 identifies distinctive features of the text spans in the text span collection 400 and in each text span in the collection 400. Loop 412 iterates for each pair of text spans. Within loop 412, step 414 determines if the pair of text spans has at least one distinctive feature in common. If they do, the pair is compared in step 416 to determine if they are duplicate or near-duplicate text spans. Loop 412 then continues with the next pair of text spans. If the pair of text spans does not have at least one distinctive feature in common, no comparison is performed, and loop 412 continues with the next pair of text spans.

This method may be used to match sentences from one document with sentences from another. This would be useful in matching sentences of a human-written summary for an original document with sentences from the original document. Similarly, in a plagiarism detector, once the method as applied to documents has found duplicate documents, the sentence version can be used to match sentences in the plagiarized copy with the corresponding sentences from the original document. Another application of sentence matching would identify changes made to a document in a word processing application where such changes need not retain the sentences, lines, or other text fragments in the original order.

Referring to Fig. 5, the present invention is utilized to find duplicate or near-duplicate images within an image collection 500. Step 510 identifies distinctive features of the images in the image collection 500 and in each image in the collection 500. The distinctive features may be sequences of at least two adjacent tiles from the images. Loop 512 iterates for each pair of images. Within loop 512, step 514 determines if the pair of images has at least one distinctive feature in common. If they do, the pair is compared in step 516 to determine if they are duplicate or near-duplicate images. Loop 512 then continues with the next pair of images. If the pair of images does not have at least one distinctive feature in common, no comparison is performed, and loop 512 continues with the next pair of images.

In a preferred embodiment of the invention according to the method illustrated in Fig. 5, the method performs canonicalization of the images by converting them to black and white and sampling them at several resolutions. As compared to the method applied to text, small overlapping tiles correspond to words and horizontal and vertical sequences to text fragments.

The method illustrated in Fig. 5 may be applied to detecting copyright infringement based on image content where the original image does not have a digital watermark. This method may also be applied to fingerprint identification or handwritten signature authentication, among other applications.

The present invention also includes an apparatus that is capable of identifying duplicate and near-duplicate documents in a large collection of documents. The apparatus includes a means for initially selecting distinctive features contained within the collection of documents, a means for subsequently identifying the distinctive features contained in each document, and a means for then comparing the distinctive features of each pair of documents having at least one distinctive feature in common to determine whether the documents are duplicate or near-duplicate documents.

Fig. 6 illustrates an embodiment of an apparatus of the present invention capable of enabling the methods of the present invention. A computer system 600 is utilized to enable the method. The computer system 600 includes a display unit 610 and an input device 612. The input device 612 may be any device capable of receiving user input, for example, a keyboard or a scanner. The computer system 600 also includes a storage device 614 for storing the document collection and a storage device 616 for storing the method according to the present invention. A processor 618 executes the method stored on storage device 616 and accesses the document collection stored on storage device 614. The processor is also capable of sending information to the display unit 610 and receiving information from the input device 612. Any type of computer system having a variety of software and hardware components which is capable of enabling the methods according to the present invention may be used, including, but not limited to, a desktop system, a laptop system, or any network system.

The present invention was implemented in accordance with the method illustrated in Figs. 3A and 3B. In the implementation, DF(x) was the number of documents containing the text "x", N was the overall number of documents, and R was a threshold on DF. Possible choices for R included a constant, a fixed percentage of N (for example five percent), the logarithm of N, or the square root of N.

A first pass over all the documents computed DF(x) for all words in the documents after converting the words to lowercase and removing punctuation from the beginning and end of the word. Optionally, a word in a particular document may be restricted from contributing to DF(x) if the word's frequency in that document falls below a user-specified threshold.

A second pass gathered the distinctive features or phrases. A phrase consisted of at least two words which occur in more than one document and in no more than R documents ($1 < DF(x) < R$). The phrases also contained glue words that occurred in at least ($N - R$) documents. The glue words could appear within a phrase, but not in the leftmost or rightmost position in the phrase. Essentially, the document was segmented at words of intermediate rarity ($R < DF(x) < R-N$) and what remained were considered distinctive phrases. Optionally, the phrases may also be segmented at the glue words to obtain additional distinctive sub-phrases, for example, "United States of America" yields "United States" upon splitting at the "of". The second pass also built a document index that mapped each document to its set of distinctive phrases and sub-phrases using a document identifier and a phrase identifier and built a phrase index that mapped from the phrases to the documents that contained them using the phrase and document identifiers. The indexes were built in a manner that ignores duplicates.

Unlike single words of low DF, the phrases were long enough to distinguish documents that happened to use the same vocabulary, but short enough to be common among duplicate documents.

A third pass iterated over the document identifiers in the document index (it is not necessary to use the actual documents once the indexes are built). For each document identifier, the document index was used to gather a list of the phrase identifiers. For each phrase identifier, the document identifiers obtained from the phrase index was iterated over to count the total number of times each document identifier occurred. Thus, for each document identifier, a list of documents that overlap with the document in at least one phrase and the number of phrases that overlap was generated. This list of document identifiers included only those documents that had at least one phrase in common with the source document in order to avoid the need to compare the source document with every other document. For each pair of documents, an overlap ratio was calculated by dividing the number of common phrases by the smaller of the number of phrases in each document. This made it possible to detect a small passage excerpted from a longer document. The overlap ratio was compared with a match percentage threshold. If it exceeded the threshold, the pair was reported as potential near-duplicates. Optionally, the results may be accepted as is or a more detailed comparison algorithm may be applied to the near-duplicate document pairs.

This implementation is rather robust since small changes to a document have little impact on the effectiveness of the method. If there are any telltale signs of the original document left, this method will find them. Moreover, the distinctive phrases do not need to appear in the same order in the duplicate documents.

The implementation is also very efficient. The first two passes are linear in N. The third pass runs in time N*P, where P is the average number of documents that overlap in at least one phrase. In the worst case P is N, but typically P is R. Note that as R increases, so does the accuracy, but the running time also increases. So, there is a trade-off between running time and accuracy. In practice, however, an acceptable level of accuracy is achieved for a running time that is linear in N. This is a significant improvement over algorithms which would require pairwise comparisons of all the documents, or at least N-squared running time.

The implementation was executed on 125 newspaper articles and their corresponding human-written summaries, for a total of 250 documents. For each pair of documents identified as near-duplicates, if the pair consisted of an article and its summary, it was counted as a correct match. Otherwise, it was counted as an incorrect match. For the purpose of the experiment, pairs consisting of a document and itself were excluded because the implementation successfully matches any document with itself. Using a minimum overlap threshold of 25% and a DF threshold of 5%, the method processed all 250 documents in 13 seconds and was able to match 232 of the 250 documents with their corresponding summary or article correctly, and none incorrectly. This represents a precision (accuracy) of 100%, a recall (coverage) of 92.8%, and an F1 score (harmonic mean of precision and recall) of 96.3%. Inspection of the results showed that in all cases where the algorithm did not find a match, the highest ranking document, although below the threshold, was the correct match.

The implementation may use different thresholds for the low frequency and glue words. Sequences of mid-range DF words where the sequence itself has low DF, may be included. Additionally, the number of words in a phrase may be factored in as a measure of the phrase's complexity in addition to rarity, for example, dividing the length of the phrase by the phrase's DF (TL/DF or log(TL)/DF). Although, this yields a preference for longer phrases, it allows longer phrases to have higher DF and, thus, be less distinctive.

It will be understood by those skilled in the art that while the foregoing description sets forth in detail preferred embodiments of the present invention, modifications, additions, and changes may be made thereto without departing from the spirit and scope of the invention. Having thus described my invention with the detail and particularity required by the Patent Laws, what is desired to be protected by Letters Patent is set forth in the following claims.